

**CLAIMS**

1. A computing device programmed with a client that can operate with a parser or  
5 generator for both text and binary mark up languages; in which the client uses a unique  
integer value that can be interpreted in an index of elements, attributes and attribute  
values needed to describe a particular type of mark-up document, the index mapping that  
unique integer value (a) to a token associated with predefined element, attribute or  
attribute value to enable a token based mark up language to be handled and also (b) to a  
10 string associated with a predefined element, attribute or attribute value to enable to  
enable a string based mark up language to be handled.
2. The device of Claim 1 in which the text mark up language is XML and the binary  
mark up language is WBXML.
3. The device of Claim 1 or 2 in which a table of mappings of each of the tokens to  
15 each of the strings is created and each mapping is given one of the unique integer values.
4. The device of Claim 3 in which two lists of unique integer values are created: one  
indexed on tokens and the other indexed on the index of the position of a string in a  
string pool table.
5. The device of any preceeding claim in which there is an extensible framework  
20 that accepts one or more mark-up language parsers and/or generators, each implemented  
as plug-ins to the framework, with different plug-ins enabling different kinds of mark up  
languages to be handled by the device.
6. The device of Claim 5 in which there is a namespace plug-in to the extensible  
framework that sets-up all the elements, attributes and attribute values for a namespace.  
25
7. The device of Claim 6 in which the index is encapsulated in the namespace plug-  
in and therefore is insulated from the client, parser and generator.
8. A method of parsing a mark-up language document, comprising the step of a  
30 client using a unique integer value that is interpreted in an index of elements, attributes

and attribute values needed to describe a particular type of mark-up document, the index mapping that unique integer value (a) to a token associated with predefined element, attribute or attribute value to enable a token based mark up language to be handled and also (b) to a string associated with a predefined element, attribute or attribute value to enable to enable a string based mark up language to be handled.

9. A method of generating a mark-up language document, comprising the step of a client using a unique integer value that is interpreted in an index of elements, attributes and attribute values needed to describe a particular type of mark-up document, the index mapping that unique integer value (a) to a token associated with predefined element, attribute or attribute value to enable a token based mark up language to be handled and also (b) to a string associated with a predefined element, attribute or attribute value to enable to enable a string based mark up language to be handled.

10. The method of Claim 8 or 9 in which the text mark up language is XML and the binary mark up language is WBXML.

11. The method of any preceding Claim 8 - 10 in which a table of mappings of each of the tokens to each of the strings is created and each mapping is given one of the unique integer values.

12. The method of any preceding Claim 8 - 11 in which two lists of unique integer values are created: one indexed on tokens and the other indexed on the index of the position of a string in a string table.

13. The method of any preceding claim 8 - 12 in which there is an extensible framework that accepts one or more mark-up language parsers and/or generators, each implemented as plug-ins to the framework, with different plug-ins enabling different kinds of mark up languages to be handled by the device.

14. The method of Claim 13 in which there is a namespace plug-in to the extensible framework that sets-up all the elements, attributes and attribute values for a namespace.

15. The method of Claim 14 in which the index is encapsulated in the namespace plug-in and therefore is insulated from the client, parser and generator.